homework6.R

```r
library(plyr)
library(e1071)
library(caret)
```

```
## Loading required package: lattice

## Loading required package: ggplot2
```

```r
library(fpc)
data <- read.table('iris.data',sep=",",header=F)
names(data)
```

```
## [1] "V1" "V2" "V3" "V4" "V5"
```

```r
set.seed(666)
names(data) <-c('sepalLength','sepalWidth','petalLength','petalWidth','category')


CVgroup <- function(k,datasize,seed){
  cvlist <- list()
  set.seed(seed)
  n <- rep(1:k,ceiling(datasize/k))[1:datasize]    #divide data into k, generate whole dataset n
  temp <- sample(n,datasize)   #shuffle n
  x <- 1:k
  dataseq <- 1:datasize
  cvlist <- lapply(x,function(x) dataseq[temp==x])  #randomly generate k random sequential dataset
  return(cvlist)
}

k <- 10
datasize <- nrow(iris)
cvlist <- CVgroup(k=k, datasize = datasize, seed=1206)

train <- iris[-cvlist[[1]],]
test <- iris[cvlist[[1]],]


nb1 <- naiveBayes(Species ~., data=train)
prediction <- predict(nb1, test)

confusionMatrix <- table(prediction,test$Species)
confusionMatrix
```

```
##
## prediction   setosa versicolor virginica
##   setosa          4          0         0
##   versicolor      0          3         0
##   virginica       0          1         7
```

```
accuracy = sum(confusionMatrix[row(confusionMatrix)==col(confusionMatrix)]) / sum(confusionMatrix)
accuracy
```

```
## [1] 0.9333333
```

```
class = sort(unique(prediction))
tp=NA
fp=NA
fn=NA
tn=NA
for(i in 1:length(class)){
  tp[i] = sum(test$Species==class[i] & prediction==class[i])
  tn[i] = sum(test$Species!=class[i] & prediction!=class[i])
  fp[i] = sum(test$Species==class[i] & prediction!=class[i])
  fn[i] = sum(test$Species!=class[i] & prediction==class[i])
}
precision = tp/(tp + fp)
recall = tp/(tp + fn)
f1score = (2*precision*recall)/(precision+recall)

names(f1score) = class
names(precision) = class
names(recall) = class

print(table(test$Species,prediction))
```

```
##             prediction
##              setosa versicolor virginica
##   setosa          4          0         0
##   versicolor      0          3         1
##   virginica       0          0         7
```

```
print('precision')
```

```
## [1] "precision"
```

```
print(precision)
```

```
##     setosa versicolor  virginica
##       1.00       0.75       1.00
```

```
print('recall')
```

```
## [1] "recall"
```

```r
print(recall)
```

```
##     setosa versicolor  virginica
##      1.000      1.000      0.875
```

```r
print('f1score')
```

```
## [1] "f1score"
```

```r
print(f1score)
```

```
##     setosa versicolor  virginica
##  1.0000000  0.8571429  0.9333333
```

```r
print('mean(f1score)')
```

```
## [1] "mean(f1score)"
```

```r
print(mean(f1score))
```

```
## [1] 0.9301587
```

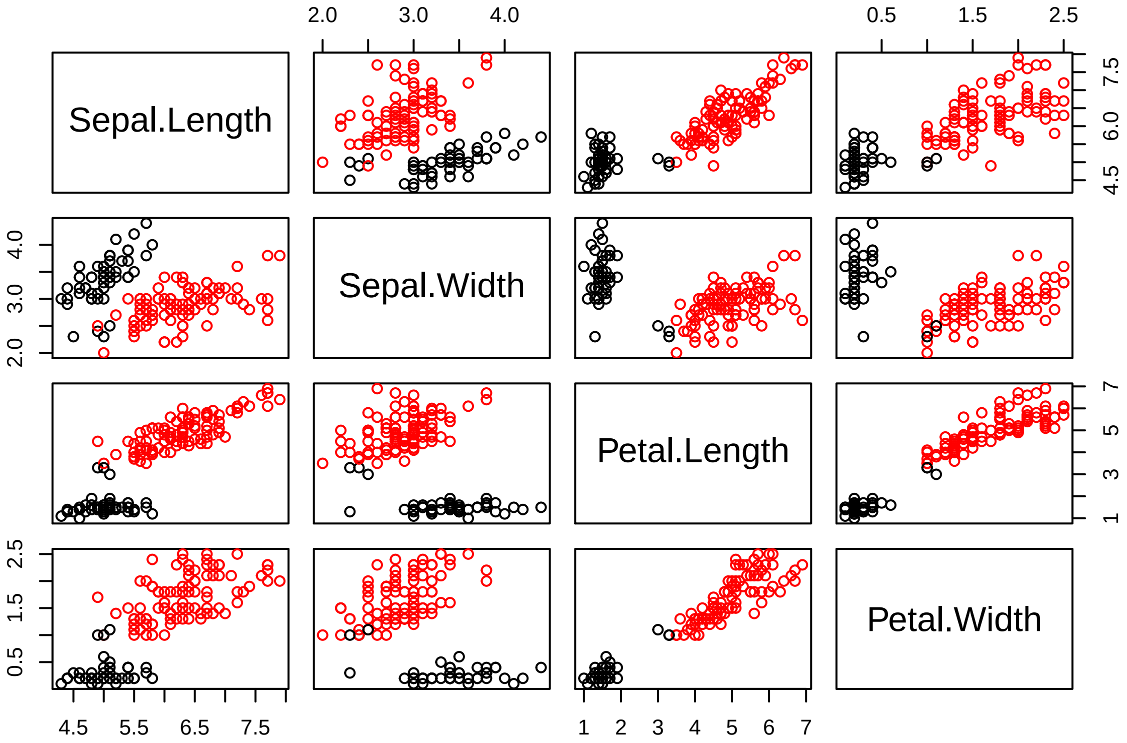
```r
#Qusetion 4


newIris<-iris[1:4]

kc<-kmeans(newIris,2)
kc$tot.withinss
```

```
## [1] 152.348
```

```r
kc$betweenss
```

```
## [1] 529.0226
```

```r
plot(newIris, col=kc$cluster)
```



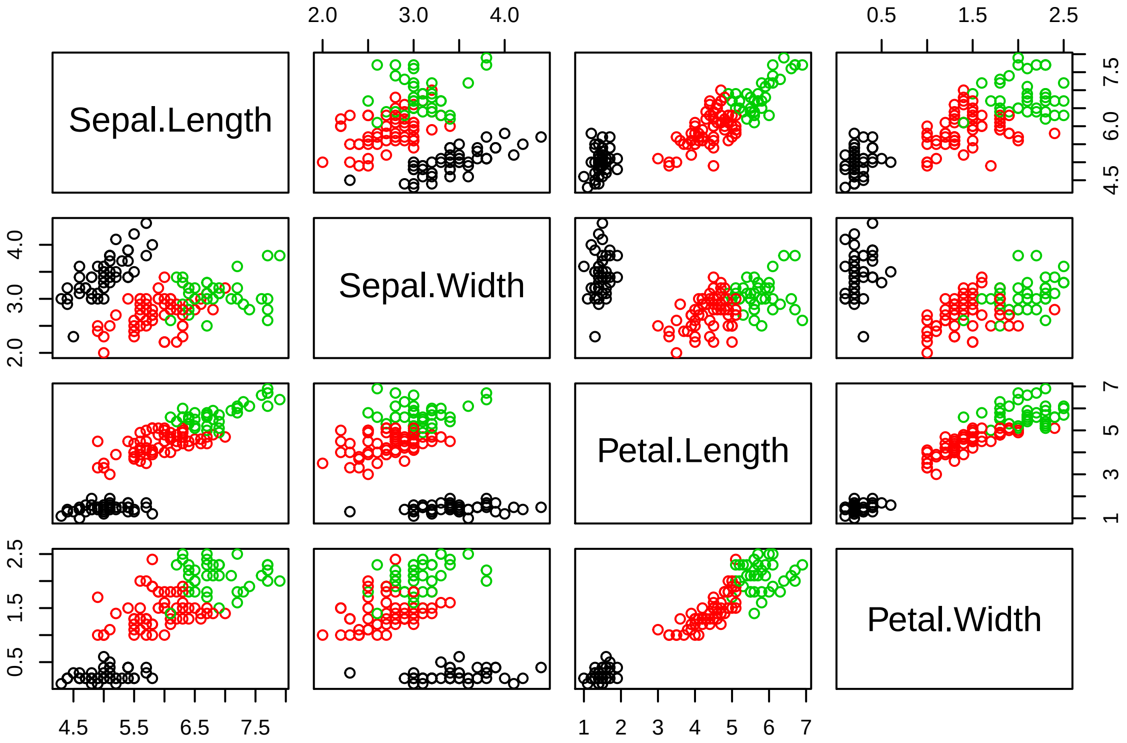
```
kc<-kmeans(newIris,3)
kc$tot.withinss
```

```
## [1] 78.85144
```

```
kc$betweenss
```

```
## [1] 602.5192
```

```
plot(newIris, col=kc$cluster)
```



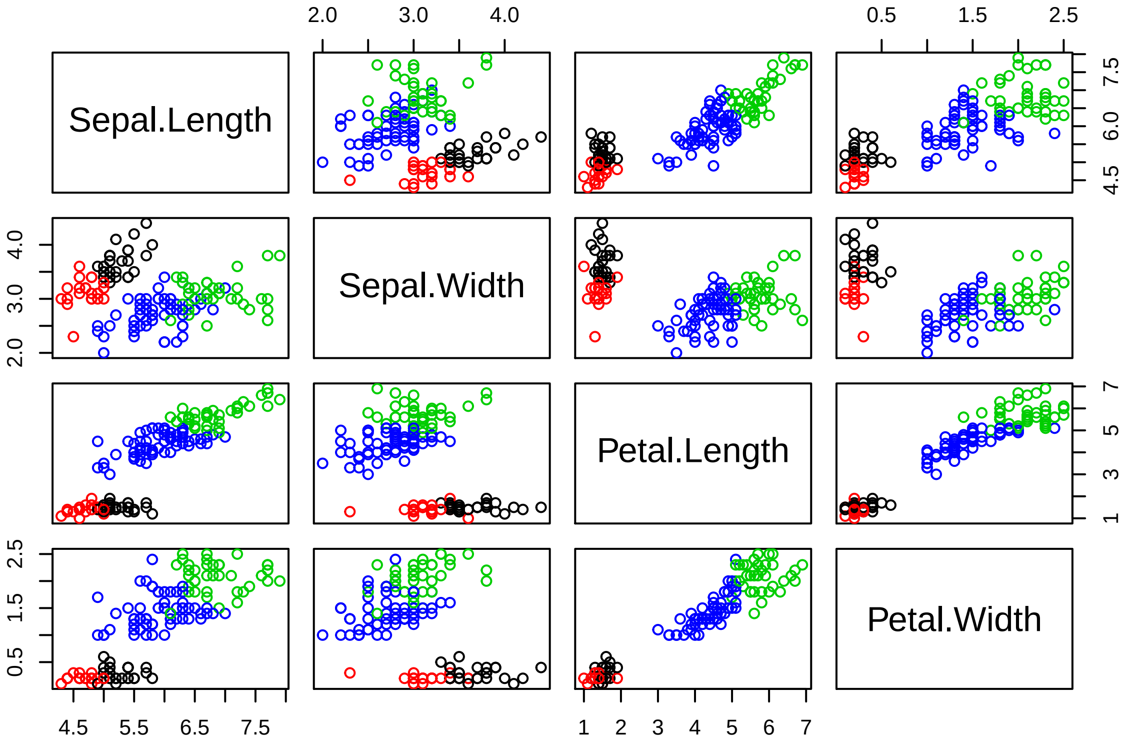
```r
kc<-kmeans(newIris,4)
kc$tot.withinss
```

```
## [1] 71.44525
```

```r
kc$betweenss
```

```
## [1] 609.9254
```

```r
plot(newIris, col=kc$cluster)
```



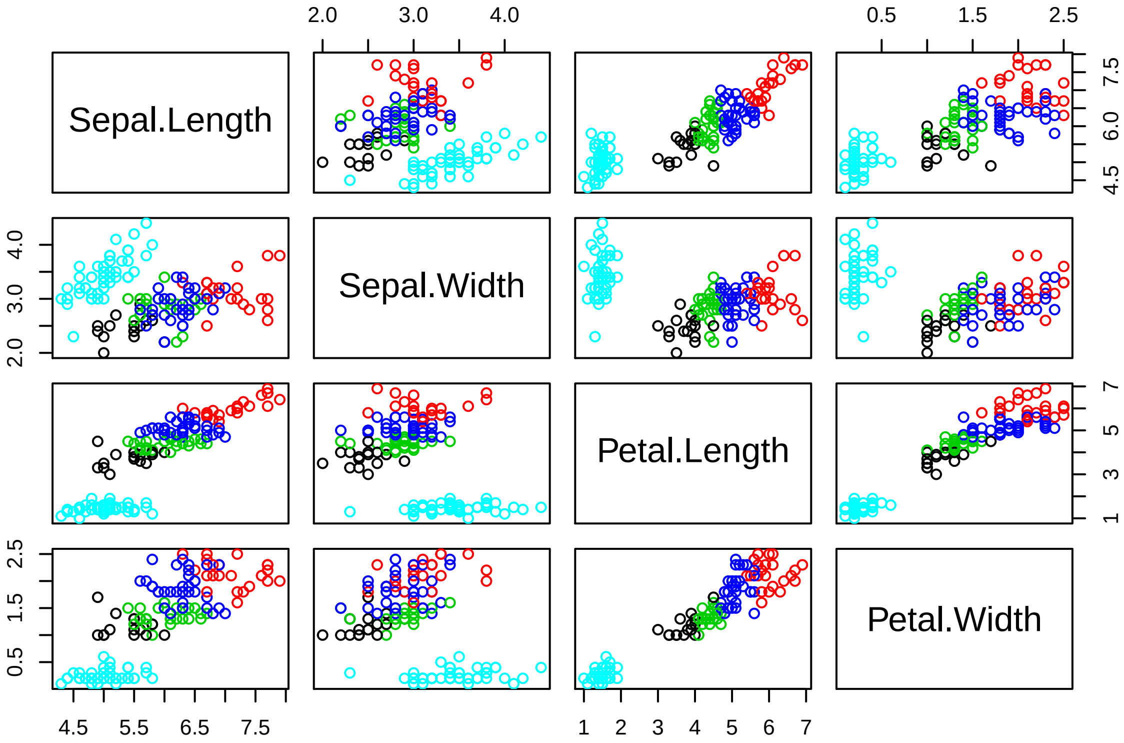
```
kc<-kmeans(newIris,5)
kc$tot.withinss
```

```
## [1] 51.08942
```

```
kc$betweenss
```

```
## [1] 630.2812
```

```
plot(newIris, col=kc$cluster)
```



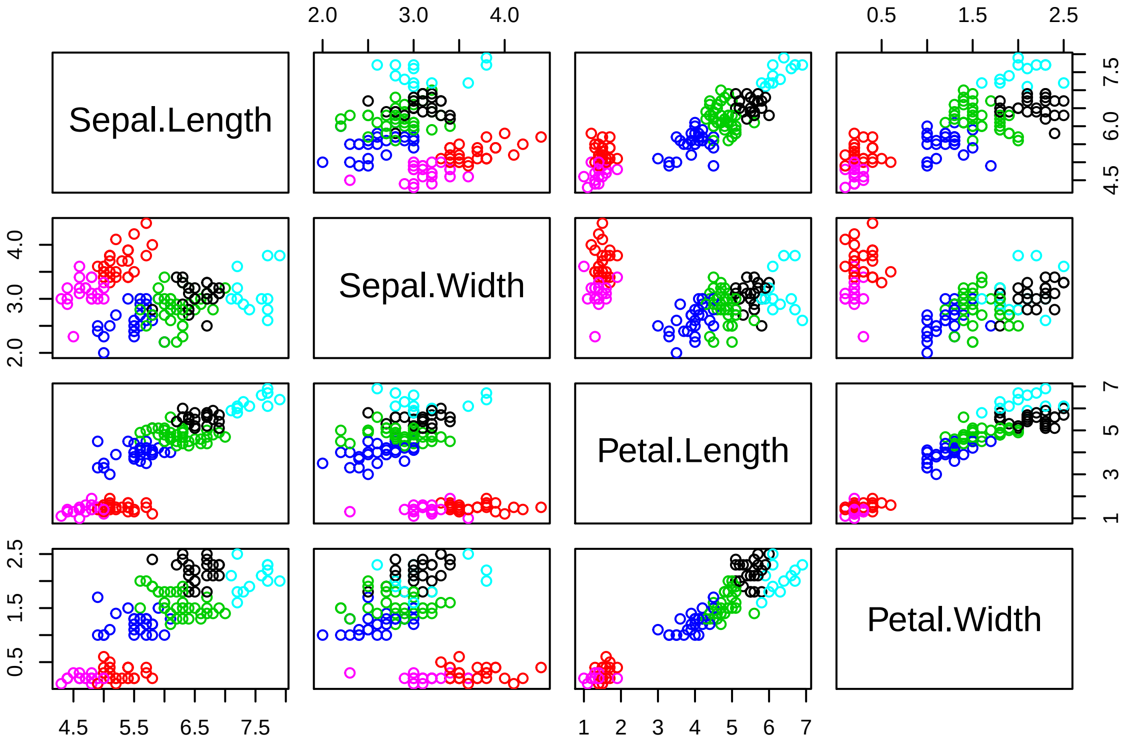
```
kc<-kmeans(newIris,6)
kc$tot.withinss
```

```
## [1] 39.05498
```

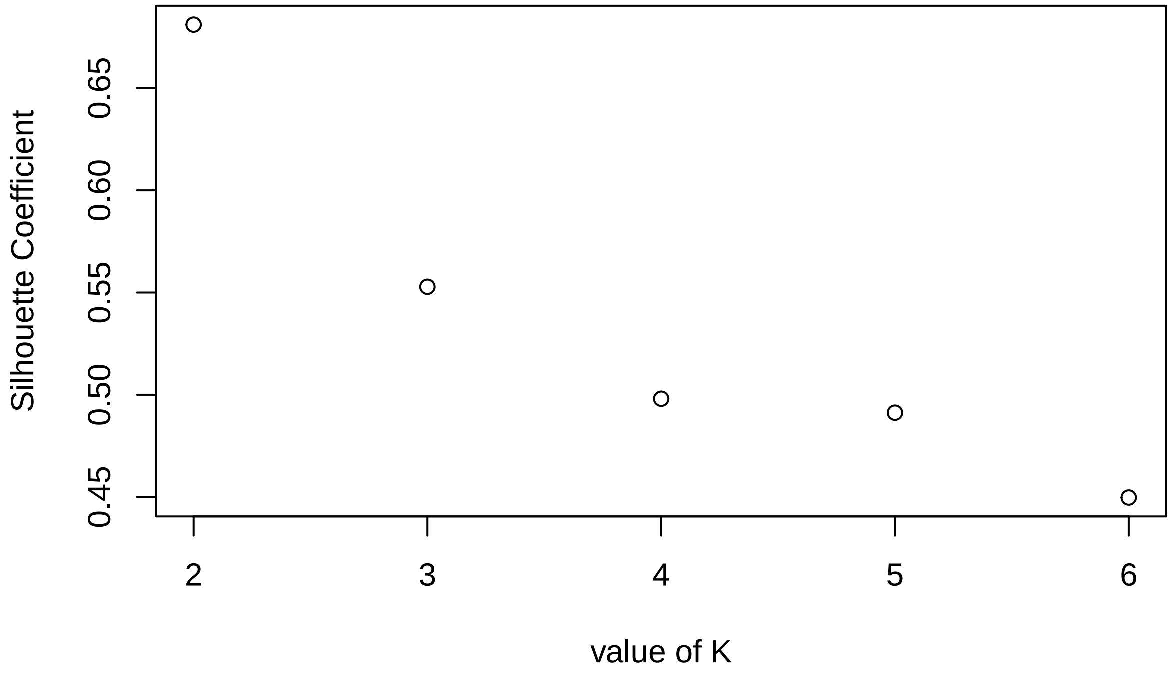
```
kc$betweenss
```

```
## [1] 642.3156
```

```
plot(newIris, col=kc$cluster)
```



```r
K <- 2:6
round <- 1
rst <- sapply(K,function(i){
  mean(sapply(1:round,function(r){
    result <- kmeans(newIris, i)
    stats <- cluster.stats(dist(newIris), result$cluster)
    stats$avg.silwidth
  }))
})

plot(K,rst,type='p', xlab='value of K',ylab='Silhouette Coefficient')
```



```
#rbind(K,rst)

kc <- kmeans(newIris,3)

confusion_matrix <- table(kc$cluster,iris$Species)
confusion_matrix
```

```
##
##     setosa versicolor virginica
##   1     17          4         0
##   2     33          0         0
##   3      0         46        50
```